

Getting Started with the Intel[®] Fortran Compiler 6.0 for Windows*

1. Overview

The Intel[®] Fortran Compiler for Windows* provides tools for Windows software developers to create applications to run on IA-32 and Itanium[™] processor-based systems. This document explains how to install the Intel Fortran Compiler for Windows, build "Hello World" for IA-32 and Itanium processor-based systems and how to get started optimizing your applications with the Intel compilers.

The Intel Fortran Compiler for Windows consists of the following:

- Intel[®] Fortran Compiler for IA-32-based applications: ifl
- Intel[®] Fortran Compiler for Itanium[™]-based applications: efl (IA32-based compiler)
- Intel[®] Fortran Itanium[™] Compiler for Itanium[™]-based applications: efl
- IA32-based Assembler to produce Itanium[™]-based applications
- Itanium[™]-based Assembler to produce Itanium[™]-based applications
- Enhanced Debugger: edb
- Fortran build tool
- Product documentation

It may be used either from a makefile, from the command line or as a plug-in to Microsoft* Visual Studio* version 6.0.

2. Installation

Administrative privileges are required in order to install the Intel Fortran Compiler for Windows.

Microsoft* Visual C++* must be installed and run at least once prior to installing the Intel[®] Fortran Compiler for IA-32-based applications. The Microsoft* Platform SDK must be installed on your development system prior to installing the Intel[®] Fortran Compiler for Itanium[™]-based applications, whether your development system is IA-32 or Itanium-based. The desired components of the Intel Fortran Compiler may then be installed from CD or from self-extracting electronic download.

The Intel® Fortran Compiler uses GlobeTrotter's* FLEXlm* electronic licensing technology. Please ensure that the license directory, default location C:\Program Files\Common Files\Intel\Licenses for both IA32 and Itanium-based systems, contains a current license file (extension .lic). A license file is included on the CD and is typically sent by Email for electronic downloads.

Any user intending to use the compiler through the Microsoft Visual Studio IDE via the Fortran Build Tool, other than the administrator who originally installed it, should first run Programs\Intel(R) Software Development Tools\ Intel(R) Fortran Compiler 6.0\Update User's Registry from the Start menu, before using the compiler for the first time. This will update the Microsoft Visual Studio registry entries for the local user.

A command window containing the correct environment for the Intel Fortran Compiler may be created from the Start menu and selecting one of Programs\Intel(R) Software Development Tools\ Intel(R) Fortran Compiler 6.0\Intel(R) Fortran for 32-bit applications or ...\Intel(R) Fortran for Itanium(TM) - based applications . To set up the environment from a regular command prompt window, you should execute C:\Program Files\Intel\Compiler60\bin\ifcvars.bat or ...efcvars.bat for the compilers for IA-32-based or Itanium-based applications respectively. The compiler can then be invoked by the commands ifl for IA-32 based applications or efl for Itanium-based applications.

Post-Installation Instructions for Itanium processors

If you did not install the Itanium-based compiler on the Itanium-based system and wish to use the IA32-based compiler to produce Itanium-based applications, you must perform the following steps in order to copy certain required Itanium-based DLLs from your IA32 development system to the Itanium-based test system:

1. Bring up a command shell on the Itanium-based system
2. Cross-mount the IA32 system partition where you installed the 6.0 compiler. For example,
net use k: \\myia32system\e\$ /user:administrator
3. Change your scope to the k drive by typing "k:" followed by a carriage return
4. Change your directory to the ia64\bin directory of the 6.0 compiler. For example,
cd \"Program Files\"Intel\compiler60\ia64\bin
5. Invoke the dll_copy.bat script by typing "dll_copy.bat" followed by a carriage return

The above steps are not necessary if you have already installed the Itanium-based compiler on the Itanium-based system, because the setup program copies the necessary DLLs by default.

NOTE: The DLL's in the Platform SDK directory "Microsoft Platform SDK\redist\PreRelease\win64" may also be required at runtime, and should be copied

into the same directory as the executable on the Itanium-based system, in order to take precedence over any versions distributed with the operating system. Some of these dlls are not distributed with the operating system and are only available from the Platform SDK.

3. Using the Intel Fortran Compiler from the command line to compile helloworld.f

The following steps show you how to compile `helloworld.f` with the Intel Fortran Compiler.

1. Create a simple `helloworld.f` program

```
PROGRAM HELLO_WORLD  
  
  PRINT *, ' Hello World! '  
  
END PROGRAM HELLO_WORLD
```

2. Open a command prompt window for the Intel Fortran Compiler from the Start menu:

Programs\Intel(R) Software Development Tools\ Intel(R) Fortran Compiler 6.0\Intel(R) Fortran for 32-bit applications or Programs\Intel(R) Software Development Tools\ Intel(R) Fortran Compiler 6.0\Intel(R) Fortran for Itanium(TM) - based applications

3. Compiling `helloworld.f` with following and it should generate "helloworld.exe".

`ifl helloworld.f` on an IA-32 system

`efl helloworld.f` on an Itanium processor-based system

4. Run the executable with "helloworld", and it should display " Hello World! ".

4. Using the Intel Fortran Compiler for IA-32-based applications from the Microsoft Visual Studio 6.0

The Microsoft Visual Studio environment does not provide a tool for compiling/building Fortran applications. To preserve customer investment in Visual Studio, Intel provides a Build Tool that associates the Intel Fortran Compiler 6.0 for 32-bit applications with Fortran files, allowing users to build IA-32 projects containing Fortran files from within Microsoft Visual Studio.

Once the Build Tool is installed, a new toolbar is available from Visual Studio that contains a blue icon with a yellow cross, as shown below, which invokes the Build Tool.



The Build Tool startup icon (the "+" on the right) and startup icon for the Enhanced Debugger (the microscope), both from Intel, are available in a new toolbar added to Visual Studio

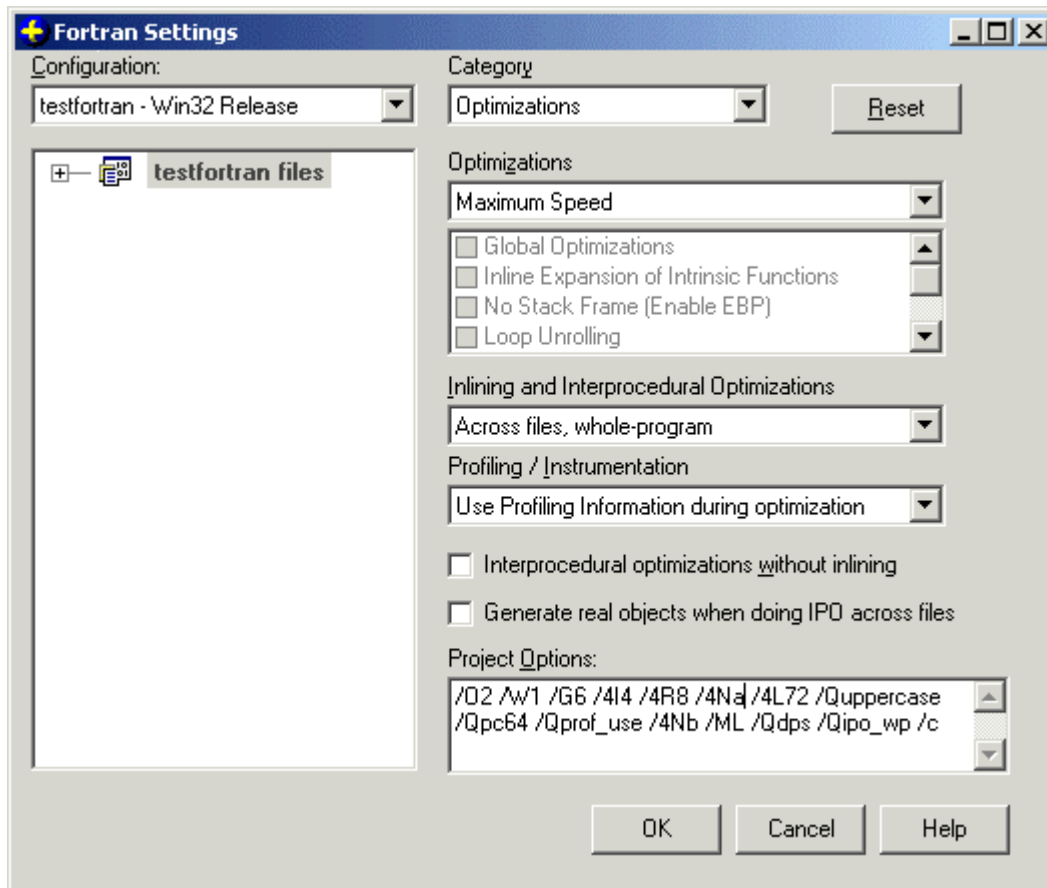
In addition to the Build Tool, the toolbar also contains an icon of a microscope that is used to invoke the Enhanced Debugger (EDB). This is a source-level symbolic debugger used for development and testing of Fortran and C/C++ applications.

Build Tool Capabilities

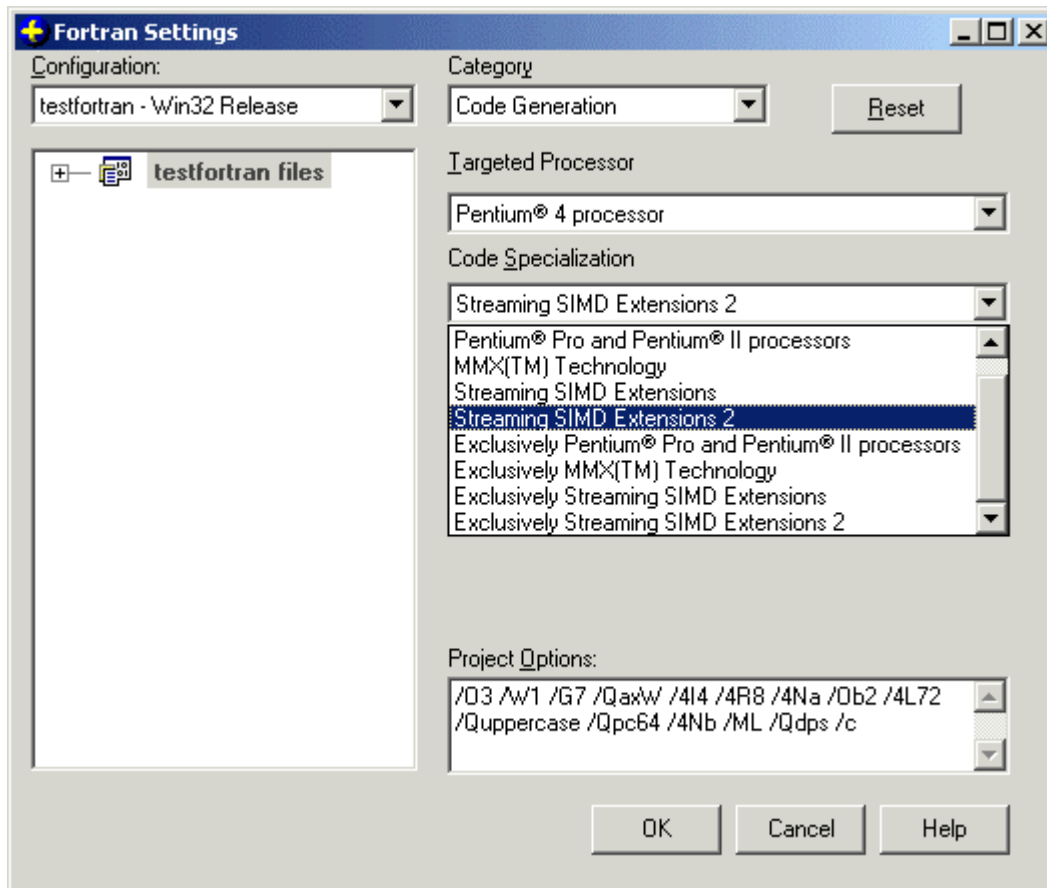
The Build Tool allows users to select Fortran compiler settings such as optimization switches at a project level or on an individual file-by-file basis. When the Build Tool is invoked, the user sees a list of the project's Fortran files and lists of compiler options per category. Currently, the compiler option categories include:

- General options
- Diagnostic options
- Language options
- Optimization options
- Floating point precision options
- Code generation options
- Output file options

The image below displays an example of the Build Tool and also depicts a sampling of optimization settings.



Another image of the Build Tool is provided below depicting code generation options and options associated with Processor Dispatch.



For help in setting up and using the Build Tool, see the section "Using Intel® Fortran Compiler within the Microsoft* Visual C++* IDE" in the chapter "Getting Started with the Intel® Fortran Compiler" of the Intel® Fortran Compiler User's Guide, accessible from the Start menu via Programs\Intel(R) Software Development Tools\ Intel(R) Fortran Compiler 6.0\Intel(R) Fortran Compiler document index .

5. Getting Started with Intel Compiler Optimizations

The Intel Fortran Compiler enables programmers to take full advantage of the advanced performance enhancement features of Intel's latest IA-32 and Itanium™ processors. These include support for

1. Streaming SIMD Extensions 2 (IA-32 only)
2. Profile-guided optimization
3. Interprocedural optimization
4. Vectorization and software pipelining
5. Processor dispatch (IA-32 only)

6. OpenMP* and auto-parallelization

These optimizations are intended for use in product-release builds of applications, not necessarily for earlier phases of application development cycles. In general, increasing the degree of optimization done by the compiler leads to an increase in compile-time and reduced debugging capability of the resulting executable. This section describes a methodology to aid in maximizing the performance achieved with the Intel Fortran Compiler using the advanced Intel optimization features.

During the application development and testing phases, you probably want to use the `/Zi/Od` switches to allow fast compile times and full debugging capabilities of your application with no optimization. To start to optimize, the default optimization `/O2` is recommended. Then `/O3` enables more advanced optimizations. Interprocedural optimization allows the compiler to optimize across different compilation units and can yield significant performance improvements. Profile guided optimization uses information from running an instrumented executable that allows the compiler to rebuild the application knowing where the majority of the computations are.

Of course, not all of the advanced optimizations are beneficial for all applications. For example, vectorization is intended for those applications that can benefit from using MMX™ technology or the Streaming SIMD Extensions 2. These are typically applications where the majority of the computation time is spent in integer (MMX technology) or floating-point loops, either single or double precision, such as those commonly found in 3D graphics applications and scientific applications.

Similarly, profile-guided optimization is most beneficial to applications where the computation is spread over several individual routines and modules; i.e., it is not confined to individual loops or kernels. Enterprise or database applications represent a class of applications that typically benefit from profile-guided optimization.

Remember to always measure the performance of your application after each optimization added to verify the benefits. The VTune™ Performance Analyzer can be a great help for measuring the performance benefits of each, as well as giving advice on further tuning opportunities. Additional information is available at <http://www.intel.com/software/products/vtune/>.

The paper, *Optimizing Applications with the Intel® C++ and Fortran Compilers*, explains how to use the Intel compilers to optimize for the Pentium® 4 and Itanium processors and is available at <http://www.intel.com/software/products/compilers/f60>. For complete information on the individual optimizations, please refer to the *Intel® Fortran Compiler User's Guide*, accessible from the Start menu via Programs\Intel(R) Software Development Tools\ Intel(R) Fortran Compiler 6.0\Intel(R) Fortran Compiler document index .

6. Uninstalling the Compiler

This may be done from the Start menu at Programs\Intel(R) Software Development Tools\ Intel(R) Fortran Compiler 6.0\Modify or Remove Intel(R) Fortran and EDB

7. Useful links and tools

Intel® Premier Support web site: Your feedback is very important to us. To receive technical support for the tools provided in this product and technical information including FAQ's and compiler updates, you need to be registered for an Intel Premier Support account on our secure web site, <https://premier.intel.com>.

1. You can register for an Intel Premier Support account at <http://support.intel.com/support/performancetools/fortran/v6/windows>. Note, if you already have access to Intel Premier Support and the “Intel(R) Fortran Compiler, Windows” product you do not need to re-register.
2. Compiler support information, including top technical issues and known issues, is available at <http://support.intel.com/support/performancetools/fortran>.
3. **Release Notes:** accessible from the Start menu via Programs\Intel(R) Software Development Tools\ Intel(R) Fortran Compiler 6.0\Intel(R) Fortran Compiler document index
4. **Compiler User's Guide:** also accessible from the Start menu via Programs\Intel(R) Software Development Tools\ Intel(R) Fortran Compiler 6.0\Intel(R) Fortran Compiler document index
5. **Package ID :** The package ID contains information about the Intel Fortran Compiler that should be supplied when submitting support issues. It may be obtained from the Start menu at Programs\Intel(R) Software Development Tools\ Registration and Support\Get Package ID

Appendix: System Requirements

IA-32 Processor System Requirements

- A system based on a Pentium®, Xeon™ or subsequent IA-32 processor (Pentium 4 processor recommended)
- 128 MB of RAM (256 MB recommended)
- 100 MB of disk space. During installation only, allow an additional 150 MB for the download and temporary intermediate files.
- 100 MB of hard disk space for the virtual memory paging file. Be sure to use at least the minimum amount of virtual memory recommended by Microsoft* Windows*.

- Microsoft Windows* 98, Windows* 98 SE, Windows* Millennium Edition, Windows* NT 4.0 with Service Pack 6, Windows* 2000, or Windows* XP.
- Microsoft Visual C++* 6.0, Professional Edition (Visual C++ Standard Edition is not supported) or Microsoft Visual C++* .NET. The Fortran build tool cannot be used as a plug-in to Microsoft Visual Studio* .NET

IA-32 Processor System Requirements for building Itanium™ processor-based applications

- A system with a 350 MHz Pentium II processor or greater
- 256 MB of RAM
- 100 MB free hard disk space (130MB if the compiler for IA32 applications is to be installed also). During installation only, allow an additional 150 MB for the download and temporary intermediate files.
- 100 MB of hard disk space for the virtual memory paging file. Be sure to use at least the minimum amount of virtual memory recommended for Microsoft Windows 2000 or Microsoft Windows XP.
- Microsoft Windows 2000 or Windows XP.
- Microsoft Platform SDK.

Itanium Processor System Requirements

- 512 MB of RAM (1 GB recommended)
- 100 MB free hard disk space. During installation only, allow an additional 100 MB for the download and temporary intermediate files.
- Microsoft Windows for Itanium processor-based systems.
- Microsoft Platform SDK.

The Enhanced Debugger does not support Windows 98, Windows 98 SE, or Windows Millennium Edition.

Copyright and Legal Information

Intel, Itanium, MMX, Pentium, VTune and Xeon are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries

* Other names and brands may be claimed as the property of others

Copyright (C) 2000, 2001, 2002 Intel Corporation. All Rights Reserved.